Projet Réseaux et Communication FMIN104

Enregistrement de rapports d'activité dans une entreprise.

SOMMAIRE

Description de l'architecture 3
Parallélisme 3
Protection des ressources partagées 3
Problèmes et solutions dans cette version 4
Bug #0001 : le numéro du port d'écoute du serveur n'est pas celui demandé 4
Manipulation des fichiers PDF 4
Amélioration de la sécurité 4
Protocoles de communication 4
Schémas algorithmiques 5
Mode d'emploi 6

Pour tester cette application, utilisez les login suivants :
login « `Admin_Test` », mot de passe « 0000 » pour les fonctionnalités spécifiques au contrôleur.
Login « `Employe_Test` », mot de passe « 0000 » pour les autres employés.

Cette application Client/Serveur permet un suivi quotidien de l'activité des employés d'une entreprise. A la demande d'un unique employé privilégié nommé « contrôleur », les employés rédigent leur rapport journalier. Celui-ci se présente sous une forme pré formatée. Ces rapports sont cumulés dans un fichier archivé au format pdf spécifique à chaque employé.

note : Pour des raisons de motivation personnelle, le contenu du fichier vuillemin.jpg a été remplacé par le logo de la Shin-Ra inc (pour plus d'info rechercher google: "shinra ff7").

Description de l'architecture

Cette solution de communication entre un programme serveur et ses clients met en œuvre les sockets Unix, les threads de service par client et les tableaux de sémaphores.

L'intégralité du programme est gérée par le serveur.

Le client se résume à une boucle :

-réception d'une requête serveur.
-saisie (si besoin) d'une réponse au clavier.
-envoie de la réponse au serveur.

L'architecture est articulée autour du dossier de ressources partagées nommé « ressources ». Ce dossier contient:

- Le fichier `employes.txt` : contient la liste des comptes utilisateurs.
  Chaque ligne est de la forme : « Login;mot de passe;num_droits ».
  Ces informations sont stockées en clair (pas de cryptage).
  Le numéro num_droits vaut 1 dans la ligne spécifique au contrôleur et 0 dans celles des autres employés.

- Les fichiers journaliers `jj_mm_aa-rapport.txt` qui contiennent la liste des employés pour lesquels un rapport est ou a été requis à la date `jj_mm_aa` selon la date système du serveur.

Parallélisme

Le parallélisme est assuré par la mise en œuvre des threads concurrents.
Le nombre de threads est cependant limité afin d'éviter tout risque de saturation du processeur.

Note : le nombre de threads maximum peut être défini au lancement du serveur (5 par défaut). Lorsque la limite est atteinte, les employés qui souhaitent se connecter sont invités à essayer plus tard.

Protection des ressources partagées

L'utilisation des sémaphores garantit un accès aux ressources partagées sans risque de conflit. 2 mécanismes distincts sont gérés par sémaphores :

- 1 sémaphore unique dit « global » qui verrouille l'accès aux variables d'état du serveur.
- 1 tableau de sémaphores dit « employés » qui verrouille les accès aux dossiers des employés pour la constitution du rapport.

Problèmes et solutions dans cette version

Bug #0001 : le numéro du port d'écoute du serveur n'est pas celui demandé

Description : Dans le cas ou le serveur est lancé sur un autre port, le port d'écoute du serveur n'est pas celui donné au lancement.
Conséquence : Ce bug provoque un changement inopiné du numéro de port utilisé réellement pour l'écoute du serveur.
Exemple : après l'exécution de la commande `./serveur 21345`, le client devra lancer la commande: ./client 127.0.0.1 24915
Solution de contournement : ce problème nous a amené à modifier les classes Sock et Sockdist pour y ajouter des classes d'affichage.

Manipulation des fichiers PDF

Pour la **création** des fichiers pdf, la présence du caractère « _ » under-score dans les logins a amené une modification du fichier `sauvegarde.cc` . Une boucle remplace les caractères under-scores dans le nom de l'employé par des espaces.

D'autre part, PDFLATEX est requis sur la création d'un nouveau fichier pdf.

A chaque consultation d'un rapport, l'appel à l'application evince est effectué. Un patch est attendu afin de garantir la lecture du fichier pdf quelque soit le reader installé sur la machine.

Amélioration de la sécurité

Un patch sera nécessaire pour assurer le cryptage des mots de passe stockés dans le fichier employes.txt avant d'envisager toute utilisation professionnelle. (md5 par exemple)

Protocoles de communication

Les communications client/serveur sont formatées suivant le principe suivant :

« Pour tout envoi de message, un accusé de réception est renvoyé afin de maintenir la synchronisation des 2 programmes (envoi et réception sont bloquants) »

Les communications sont de 4 types:

0: affichage simple de message chez le client

1: affichage d'une question avec réponse au format chaîne de caractère
- le client se contente de renvoyer la chaîne de caractère, il ne regarde jamais le contenu.

2: affichage d'une question dont la réponse est au format Oui/Non
- le client bouclera sur la saisie au clavier jusqu'à avoir pour réponse y ou n.

3: demande de clôture, 2 types:
- simple (le client ferme la connexion)
- double (le serveur se ferme lui aussi, et le client fait une 2ème demande de connexion pour débloquer le listen du serveur)

4: téléchargement du pdf par le client
- ouverture du fichier pdf avec le programme "evince".

Schémas algorithmiques

Serveur

```
Créer sémaphore global semID et initialiser sa valeur à 1 ;
Calculer le nombre d'employés nbEmp selon le contenu du fichier employes.txt ;
Initialiser le tableau des sémaphores empSemId ;
Créer la boite réseau publique BRpublique ;
Mettre BRpublique en écoute ;

TANT QUE (1) {
        V(semID);
        descBrTmp=accept(descBrPub) ;
        P(semId) ;
        si (connexion){
                V(semID) ;
                creation nouveau thread;
                P(semId) ;
        }
} FIN TANT QUE
V(semId) ;
pthread_join;
fermer BRpublique ;
Supprimer SemID ;
Supprimer empSemId ;
```

Client

```
Créer BR privée BrCli
        Connexion BrCli;
        Si connexion reussie à BrCli{
                Recevoir
                Envoyer accusé de réception
                Tant que (finConnexion){
                                finConnexion=respRequete(descBrCli);
                        }
                                si (finConnexion==2){
                                Créer BR privée brClose
                                Si connexion reussie à brClose
                                        recv(descBrCli,c,2,0);
                                        send(descBrCli,(char*)"ack",4,0);
                                }
                        }
                }sinon{
                        printf("Pb de connexion avec le serveur, essayez plus tard.\n");
                }
                Fermer BrCli;
        }
        sinon afficher "Connexion impossible" ;
```

Mode d'emploi

1) Lancer le serveur par la commande. /`serveur` sur une machine centrale. Le port 21345 doit être accessible. Il est possible de modifier les numéros de port d'écoute du serveur ainsi que le nombre maximum de clients avec la syntaxe suivante:

```
./Serveur [port d'écoute] [nombre maximum de threads secondaires]
```

2) Lancer un ou plusieurs clients avec la syntaxe suivante:

```
./Client [adresse IP du serveur] [port d'écoute du serveur]
```

3) Se connecter en qualité de contrôleur via le login « Admin_Test » pour effectuer une opération privilégiée :
[1] Fermer le serveur. (Ceci déconnectera tous les utilisateurs).
[2] Consulter la liste des rapports requis.
[3] Envoyer une demande de rapport en précisant le login de l'employé.
[4] Ouvrir un rapport en choisissant le login de l'employé.

4) Se connecter en qualité d'employé via le login « Employe_Test » pour écrire un rapport (si celui a été demandé ce jour par le contrôleur) ou consulter le rapport archivé.

Procédure pour ajouter un nouvel utilisateur :

1) Ajouter une ligne de la forme « Nouveau_Login;mot de passe;num_droits » dans le fichier ressources/employes.txt. (num_droits=1 pour un contrôleur et 0 pour un employé)
2) Créer le dossier « Nouveau_Login » et y ajouter les fichiers suivants : temp.aux, temp.log, temp.pdf, temp.tex.